UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE
CAMPUS ARARANGUÁ

Valter Savi Junior

ARQUITETURA DE SENSORIAMENTO AUTONÔMICA PARA ABRIGOS DE CULTIVO

Araranguá

2017

Valter Savi Junior

ARQUITETURA DE SENSORIAMENTO AUTONÔMICA PARA ABRIGOS DE CULTIVO

Trabalho de Conclusão de Curso submetido à Universidade Federal de Santa Catarina, como parte dos requisitos necessários para a obtenção do Grau de Bacharel em Engenharia de Computação.
Orientador: Prof. Anderson Luiz Fernandes Perez, Dr.

Araranguá, Dezembro de 2017.

Valter Savi Junior

ARQUITETURA DE SENSORIAMENTO AUTONÔMICA PARA ABRIGOS DE CULTIVO

Trabalho de Conclusão de curso submetido à Universidade Federal de Santa Catarina para a obtenção do Grau de Bacharel em Engenharia de Computação. Orientador: Prof. Anderson Luiz Fernandes Perez, Dr.

Araranguá

2017

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Savi Júnior, Valter
Arquitetura de sensoriamento autonômica para abrigos de cultivo / Valter Savi Júnior ; orientador, Anderson Luiz Fernandes Perez, 2017.
66 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Campus Araranguá, Graduação em Engenharia de Computação, Araranguá, 2017.

Inclui referências.

1. Engenharia de Computação. 2. Abrigos de cultivos. 3. Computação autonômica. 4. Internet das coisas. I. Fernandes Perez, Anderson Luiz . II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Computação. III. Título.

Valter Savi Junior

ARQUITETURA DE SENSORIAMENTO AUTONÔMICA PARA ABRIGOS DE CULTIVO

Este Trabalho de Conclusão de curso foi julgado aprovado para a obtenção do Título de "Bacharel em Engenharia de Computação", e aprovado em sua forma final pela Universidade Federal de Santa Catarina.

Araranguá, 05 de Dezembro 2017.

Eliane Pozzebon:77921313949
Assinado de forma digital por Eliane Pozzebon:77921313949
Dados: 2017.12.13 16:51:17 -02'00'

Profª Eliane Pozzebon, Dra.
Coordenadora

Banca Examinadora:

Anderson Luiz Fernandes Perez:02080345958

Assinado de forma digital por Anderson Luiz Fernandes Perez:02080345958
Dados: 2017.12.13 17:17:48 -02'00'

Prof. Anderson Luiz Fernandes Perez, Dr.

Alexandre Leopoldo Goncalves:78674034934

Digitally signed by Alexandre Leopoldo Goncalves:78674034934
DN: CN=Alexandre Leopoldo Goncalves:78674034934, OU=UFSC - Universidade Federal de Santa Catarina, O=ICPEdu
Reason: I attest to the accuracy and integrity of this document
Location: Araranguá/UFSC
Date: 2017-12-13 14:17:45

Prof. Alexandre Leopoldo Gonçalves, Dr.

Fabio Rodrigues de La Rocha:96300752020

Digitally signed by Fabio Rodrigues de La Rocha:96300752020
Date: 2017.12.13 16:38:36 -02'00'

Prof. Fábio Rodrigues De la Rocha, Dr.

Este trabalho é dedicado à minha família,
meus queridos amigos e namorada

AGRADECIMENTOS

Este trabalho é dedicado aos meus queridos colegas que fizeram minha jornada na universidade ser algo extraordinário e que se tornaram grandes amigos. Aos meus queridos pais, que me proporcionaram tantas oportunidades e sempre acreditaram em mim. A minha namorada Anne Dias a qual sempre esteve me apoiando. E, finalmente, e não menos importante, ao Professor Anderson Luiz Fernandes Perez que acreditou em meu potencial e sempre esteve presente para que este trabalho fosse concluído. Também agradeço a todos os integrantes do Laboratório de Automação e Robótica Móvel - LARM os quais sem eles talvez este trabalho não seria possível ser finalizado.

RESUMO

O cultivo em ambiente protegido vem aumentando em escala mundial, devido aos benefícios gerados pelo microclima criado em seu interior, que permite prover um ambiente, se manejado de forma correta, próspero para o cultivo. Este microclima, apesar de proporcionar benefícios é tido como um grande desafio, devido ao fato de necessitar um correto manejo do ambiente. Caso contrário, pode gerar um clima desfavorável colocando a cultura em perigo. Já existem sistemas de controle destes ambientes, porém ainda necessitam forte monitoramento dos equipamentos, para que o sistema opere adequadamente. Este trabalho tem como proposta utilizar o conceito de Internet das Coisas (IoT do inglês *Internet of Things*) para prover uma arquitetura de sensoriamento de um abrigo de cultivo, utilizando módulos sensores como objetos/coisas que podem promover serviços para o sistema de controle. Além dos conceitos da IoT o trabalho também propõe o uso da computação autonômica para tornar o sensoriamento do ambiente mais autônomo e seguro do ponto de vista do funcionamento dos equipamentos de monitoramento. O sistema proposto foi avaliado em diferentes cenários. Com os resultados obtidos foi possível demonstrar as potencialidades do sistema.

Palavras-chave: Internet das Coisas, Computação Autonômica, Arquitetura, Redes autônomas, Abrigos de cultivo

ABSTRACT

Growing in a protected environment is increasing worldwide, due to the benefits generated by the microclimate created in its interior, which allows to provide an environment, if managed correctly, prosperous for cultivation. This microclimate, although providing benefits is considered a great challenge, due to the fact that it needs a correct management of the environment. Otherwise it can generate an unfavorable climate putting the culture in danger. There are already systems to control these environments, but they still require a strong monitoring of the equipment, so that the system operates correctly. This work intends to use the Internet of Things (IoT) concept to provide an environment sensing architecture, using sensor modules as objects / things that can promote services for the control of the environment. In addition to the IoT concepts the paper also proposes the use of autonomic computing to make the environment sensing more autonomous and safe from the point of view of the operation of the monitoring equipment. The proposed system was evaluated in different scenarios. With the results obtained it was possible to demonstrate the potential of the system.

Keywords: Internet of Things, Autonomic Computer, Architecture, Autonomous Networks, crop shelter

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

RFID	*Radio-Frequency IDentification*
IERC	*European Research Cluster on the Internet of Things*
ITU	*International Telecommunications Union*
IoT	*Internet of Things*
TIC	Tecnologia da Informação e Comunicação
IP	*Internet Protocol*
EPC	*Eletronic Product Code*
uCode	Codificação Ubíqua
GPS	*Global Positioning System*
RTOS	*Real Time Operating System*
ID	Identificação
WiFi	*Wireless Fidelity*
NFC	*Near Field Communication*
MQTT	*Message Queuing Telemetry Transport*

SUMÁRIO

1 INTRODUÇÃO

O cultivo em ambiente protegido vem aumentando em escala mundial, sendo este geralmente feito em estufas cobertas por um material translúcido, onde em seu interior é possível gerar um microclima favorável para a cultura, proporcionando benefícios jamais alcançados no cultivo tradicional (ABREU; BASTOS et al., 2015).

O cultivo em ambientes protegidos permite que o agricultor obtenha maior eficiência no controle de pragas, redução da sazonalidade, melhor proveito dos recursos como CO_2, água e radiação solar, trazendo assim uma melhor qualidade para a cultura (GUISELINI et al., 2010).

No Brasil, esta técnica foi empregada nos anos 70, crescendo de maneira abrupta nos anos 80. Apesar da rápida popularização no país, não há dados precisos sobre a área de cultivo protegido no Brasil (VIDA et al., 2004). As informações técnicas sobre o desempenho das plantas neste âmbito provenientes de experimentos e pesquisas, demonstram que as culturas cultivadas nos abrigos possuem melhor desempenho comparadas com o cultivo tradicional (ABRAH; RODRIGUES; PAGIUCA, 2014).

O grande benefício dos ambientes protegidos, ou seja, o microclima, é também seu grande desafio. Gerenciar todas as variáveis como temperatura, umidade, luz solar, radiação, nutrientes, ventilação, etc, é algo muito delicado, erros no manejo do ambiente podem ocasionar um efeito contrário, isto é, gerar um microclima favorável para o surgimento de pragas. Além das pragas, a falha no manejo pode tornar o microclima inóspito para a cultura, colocando em risco todo o cultivo (VIDA et al., 2004; PURQUERIO; TIVELLI, 2006).

Atualmente o gerenciamento dos ambientes protegidos são todos efetuados de maneira semi autônoma, existem sistemas e ferramentas que auxiliam no controle e automação do abrigo, como sistemas de acionamento dos equipamentos (exaustores, irrigadores, sombrites, etc) para a aquisição de informações do ambiente como temperatura e umidade. Porém, a responsabilidade do gerenciamento dessas ferramentas e a correta utilização das mesmas, é de responsabilidade do agricultor. Este deve estar atento a todas as variáveis importantes para a cultura tornando assim o abrigo fortemente dependente do monitoramento do agricultor (JUNIOR, 2011; UZOCHUKWU et al., 2015).

Devido a adversidade no gerenciamento do ambiente, algumas abordagens para a automação dos abrigos de cultivos foram desenvolvidas, utilizando-se de recursos computacionais e tecnologias embar-

cadas (MARANGONI; SOUZA; MOREIRA, 2014; ABREU; BASTOS et al., 2015). De maneira geral, há o monitoramento do ambiente através de sensores, que transmitem os dados para uma central, podendo ser um microcontrolador onde os dados são interpretados e, com base nas informações obtidas o microcontrolador atua no ambiente. Este contexto de sensoriamento do ambiente, onde há a troca de informações entre equipamentos, "coisas" e objetos, está relacionado ao conceito de Internet das Coisas do (inglês IoT - *Internet of Things*).

A IoT é um conceito/paradigma que considera a comunicação e presença pervasiva de objetos que interagem e cooperam entre si em prol de um objetivo comum (FRIESS, 2013). No contexto dos ambientes protegidos este objetivo pode ser o controle do abrigo, principalmente para tarefas simples, como a irrigação, controle de temperatura e umidade, ou tarefas ainda mais complexas como identificar pragas, ou qual o tipo de cultivo e em qual estágio de desenvolvimento encontram-se a cultura (JAMES et al., 2014).

No entanto, neste contexto, ainda é necessário um gerenciamento, pois o sistema pode apresentar falhas, como sensores danificados, falha de comunicação entre serviços ou até mesmo ataques maliciosos. Em vista deste gerenciamento, a IBM, em 2001 notou a necessidade de sistemas auto-gerenciados, assim propôs uma técnica chamada computação autonômica. Esta técnica se apoia em quatro pilares denominados, propriedades auto's, quais sejam: auto-cura, auto-configuração, auto-otimização e auto-proteção (HORN, 2001).

Assim este trabalho visa integrar o conceito da IoT com as propriedades da computação autonômica de auto-cura e de auto-configuração, no desenvolvimento de um sistema de sensoreamento autônomo para ambientes protegidos.

1.1 MOTIVAÇÃO E JUSTIFICATIVA

Em 2001, o número de aparelhos conectados na internet ultrapassou o volume de pessoas em escala mundial, e espera-se que em 2020 o número de objetos conectados esteja entre os valores de 26 a 50 bilhões (JAMES et al., 2014). Além do volume dos aparelhos, a tecnologia empregada também evoluiu, possibilitando a criação de dispositivos inteligentes (GUBBI et al., 2013).

O termo *Internet of Things*, foi escrito por Kevin Ashton em 1999 perante uma apresentação realizada na empresa Procter & Gamble (ASHTON, 2009), onde propunha que os dispositivos deviam se re-

conhecer e trocar informações entre si no escopo empresarial. Com os conceitos da computação ubíqua, a IoT se consolidou como realidade, com aplicações em inúmeras áreas, como *smart cities*, *health care*, automação industrial, agricultura, etc (VERMESAN; FRIESS, 2014).

Mesmo sendo uma grande revolução, e um imenso avanço tecnológico, a IoT ainda encontra muitos desafios como, qualidade de serviço. Exemplo, demora nas respostas dos serviços devido ao grande número de dados trafegando na rede (GUBBI et al., 2013). Este número volumoso de dados é oriundo da vasta quantidade de dispositivos presentes, o que leva também a mais um desafio que é o gerenciamento destes dispositivos.

Em 2001, a IBM propôs um sistema auto-gerenciável, devido a grande complexidade dos sistemas e a dificuldade em gerenciá-los, este seria referenciado como sistema autonômico, onde o gerenciamento ocorre de maneira autônoma, sem a necessidade ou com o mínimo de intervenção humana (KEPHART; CHESS, 2003).

Alguns trabalhos já incluem a necessidade de características autonômicas na IoT como os trabalhos apresentados por Friess (2013), Vermesan e Friess (2014), Nunes, Zhang e Silva (2015).

Este trabalho visa a integração dos conceitos de IoT com base em algumas propriedades da computação autonômica para ser aplicado em ambiente de sensoriamento como o cultivo protegido, onde há necessidade de um monitoramento constante devido a sensibilidade do microclima criado nos ambientes (PURQUERIO; TIVELLI, 2006).

1.2 OBJETIVOS

Esta seção apresenta o objetivo geral e os objetivos específicos deste trabalho.

1.2.1 Objetivo Geral

Desenvolver uma arquitetura de sensoriamento em um abrigo de cultivos baseada nos conceitos de IoT e nas propriedades de auto-cura e de auto-configuração da computação autonômica.

1.2.2 Objetivos Específicos

1. Propor uma arquitetura de sensoreamento para abrigos de cultivo;
2. Criar um ambiente que simule um abrigo de cultivo para embarcar o sistema de sensoriamento proposto em (1);
3. Avaliar o sistema proposto com base nos conceitos de auto-cura e de auto-configuração da computação autonômica;
4. Analisar os resultados obtidos em (3).

1.3 ORGANIZAÇÃO DO TRABALHO

Além desta introdução este trabalho está organizado em mais 5 (cinco) capítulos que abordam os seguintes temas:

- O **Capítulo 2** aborda os principais conceitos a respeito da Internet das Coisas (IoT). São definidas as principais arquiteturas e a infraestrutura para implantação de soluções baseados em IoT.
- O **Capítulo 3** aborda a definição da computação autonômica, suas propriedades, as propriedades auto's e a arquitetura base para outras arquiteturas de computação autonômica.
- No **Capítulo 4** é descrita a arquitetura autonômica para redes de sensoriamento proposta neste trabalho e como ela está disposta em um ambiente de cultivo. Será apresentado a arquitetura da rede, características e seus módulos de hardware e de software.
- No **Capítulo 5** é descrita os experimentos e os resultados obtidos com a arquitetura de sensoriamento autonômica proposta. Inicialmente o capítulo descreve a metodologia de avaliação empregada e em seguida os resultados obtidos com os experimentos realizados.
- O **Capítulo 6** apresenta as considerações finais deste trabalho e algumas propostas para trabalhos futuros.

2 INTERNET DAS COISAS

Este capítulo aborda os principais conceitos a respeito da Internet das Coisas (IoT). São definidas as principais arquiteturas e a infraestrutura para implantação de soluções baseados em IoT.

2.1 FUNDAMENTOS DE IOT

IoT é um conceito/paradigma que considera a comunicação e presença pervasiva de objetos/coisas que interagem e cooperam entre si através de uma comunicação sem ou com fio, junto com um esquema de endereçamento único na rede de comunicação, criando serviços e aplicações que buscam objetivos comuns. Neste contexto tem-se a ideia de um mundo inteligente onde o real, digital e virtual convergem para a criação de meios como cidades, transportes, agricultura, entre outros. A meta da IoT é permitir que os objetos se conectem na rede em qualquer hora, lugar e com qualquer um, seja objeto ou não, utilizando algum caminho da rede ou serviço (VERMESAN; FRIESS, 2014).

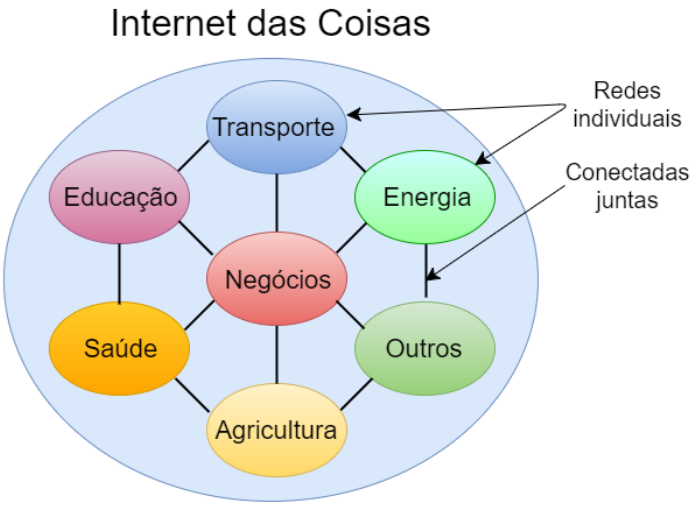
IoT é a nova revolução da internet onde cada objeto pode conectar-se na rede, reconhecer coisas e ser reconhecido por outros, atuando no ambiente de maneira inteligente, devido ao fato de poderem se comunicar e disponibilizar suas informações, ou adquirir conhecimento através de sensores ou minerando dados de outros objetos na rede (FRIESS, 2013).

A união dos protocolos de comunicação sem fio, melhor aquisição de dados do ambientes (aperfeiçoamento de sensores), redução do custo de processadores e microcontroladores, o mercado de startups e grandes empresas desenvolvendo os softwares necessários de gestão e aplicações, deram realidade a este conceito de revolução. Em 2001 o número de aparelhos conectados na internet ultrapassou o volume de pessoas em escala mundial, espera-se que em 2020 o número de objetos conectados esteja na faixa de 26 a 50 bilhões. Para cada PC ou Smartphone conectados na rede existirão por volta de 5 a 10 outros objetos que estarão aptos de fábricas à conexão com a internet. Isso inclui qualquer tipo de produto eletrônico, carros, máquinas industrias, ferramentas e muitos outros que ainda não foram inventados (JAMES et al., 2014).

Nos dias atuais tem-se redes únicas, onde objetos trocam informações e agem de maneira autônoma ou semiautônoma, como os prédios comerciais e residenciais que possuem sistemas de controle para

temperatura, ventilação, iluminação, controle de presença, etc. Nestes sistemas, as redes são individuais, ou seja, não há comunicação com outras redes fora de seu escopo. A IoT tem como objetivo interconectar as redes individuais, formando então uma redes de redes (FRIESS, 2013). A Figura 1 ilustra uma visão da IoT onde essas redes únicas estão interconectas

Figura 1 – Internet das coisas vista como uma rede de redes



Fonte: Adapatado de Cisco IBSG, April 2011

A IoT é um conceito que envolve muitas áreas, considerando o contexto e a tecnologia necessária desde sensores, sistemas de comunicação, acumulo de dados, pré-processamento e geração de serviços, não é simples definir de maneira não ambígua o que é IoT. Neste contexto a IoT foi definida pela ITU (*International Telecommunications Union*) e IERC (*European Research Cluster on the Internet of Things*) como uma infra-estrutura de rede global dinâmica com capacidade de se autoconfigurar baseada nos padrões e protocolos de comunicação interoperáveis, onde as "coisas" tanto virtuais como físicas tem identificadores, atributos físicos e personalidades latentes, utilizam interfaces inteligentes e são perfeitamente integradas na rede de informação (VERMESAN; FRIESS, 2014).

2.1.1 Principais características da IoT

As principais características da IoT estão definidas abaixo:

- Interconectividade dos objetos com a informação e infraestrutura global.
- Coisas relacionadas a serviços, ou seja, os objetos conectados podem disponibilizar serviços na rede.
- Heterogeneidade, os dispositivos conectados na rede, em suma, são heterogêneos, isto é, possuem hardwares e/ou softwares diferentes, e mesmo assim são capazes de interagir entre si.
- O estado de um dispositivo na rede pode variar com o tempo e situação, podendo estar em dormência ou ativo, até mesmo desconectado da rede, ou em lugares diferentes.
- O número de coisas/objetos conectados na rede é de tamanha proporção que geram um volume muito considerável de dados na rede.

A IoT não é uma simples tecnologia, é um conceito onde todas as "coisas"estão conectadas e capacitadas para prover soluções baseadas na integração das informações dos objetos conectados na rede, os quais captam, processam e armazenam e/ou transmitem os dados. Assim, a rápida convergência de tecnologia de comunicação esta tomando lugar em três camadas da inovação da tecnologia: nuvem, dados e comunicação sem ou com fio e dispositivos (DUTTA; BILBAO-OSORIO, 2012).

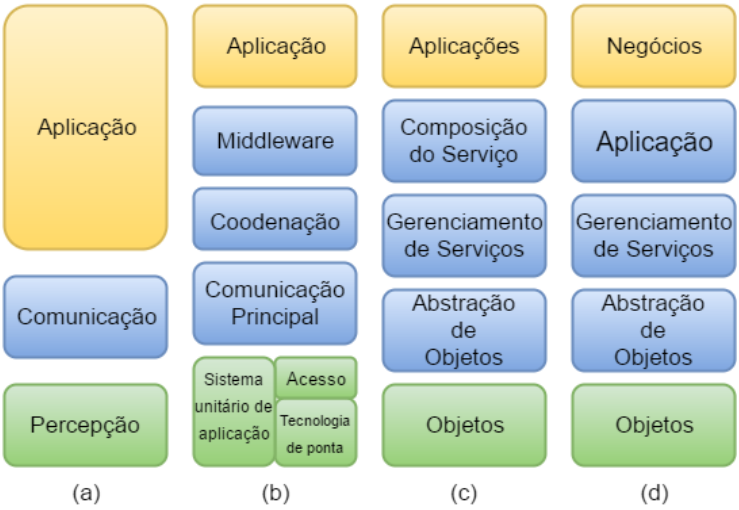
2.2 ARQUITETURA

A IoT deve ser capaz de conectar inúmeros objetos heterogêneos pela rede de comunicação. Para tal, necessita-se de uma arquitetura flexível. Apesar de existirem várias propostas, ainda não há modelo de referência devido a rápido evolução das arquiteturas (ARORA et al., 2017). Um dos projetos de arquitetura para IoT, IoT-A tem com foco as análises das necessidades da indústria e de pesquisadores (AL-FUQAHA et al., 2015).

Segundo Al-Fuqaha et al. (2015), entre as diversas arquiteturas, existe um modelo básico de três camadas para a IoT, sendo elas: aplicação, comunicação e percepção (do inglês: *Application, Network*

e Perception). Nos modelos mais recentes apresentados na literatura, tem-se a adição de mais algumas camadas de abstração, alcançando um total de cinco, como ilustra a Figura 2.

Figura 2 – Arquitetura IoT. (a) Três camadas. (b) Beaseada em middleware. (c) Arquitetura orientada a serviços (d) Cinco camadas



Adpatado de Al-Fuqaha et al. (2015)

A camada de percepção ou objetos (dispositivos) é composta por sensores e atuadores, os quais são responsáveis pela aquisição de dados e atuação no ambiente. Segundo (YUN; YUXIN, 2010; YANG et al., 2011), esta camada tem função análoga aos cinco sentidos (olfato, tato, audição, paladar e visão), no sentido de obter dados e informações do ambiente inseridos como temperatura, umidade, movimento, etc. Nesta camada as informações retiradas são digitalizadas e transferida para a próxima camada. É importante ressaltar que há heterogeneidade dos dispositivos que compõe a camada, necessitando padronização de mecanismos *plug and play* (AL-FUQAHA et al., 2015).

A camada de abstração de objetos transfere os dados recebidos da camada de objetos para a camada de gerenciamento de serviços através de canais seguros. Podendo utilizar diversas tecnologias para a transmissão, por exemplo, RFID, 3G, WiFi, Bluetooh, entre outras

(AL-FUQAHA et al., 2015). Segundo (YANG et al., 2011), outras demais funções como computação em nuvem também fazem parte desta camada.

A camada de Gerenciamento de Serviços, ou camada *Middleware* é responsável por parear as requisições de serviços com os serviços disponibilizados na rede. Nesta camada há abstração dos hardware permitindo o desenvolvimento de aplicações de IoT, sem a necessidade de plataformas de hardware específicos, e também toma decisões com base no processamento dos dados obtidos (AL-FUQAHA et al., 2015).

A camada de aplicação provê os serviços requisitados, como as informações obtidas pelos sensores. Tem a capacidade de prover serviços inteligentes de alta qualidade, englobando um vasto mercado, como *smart homes*, automação industrial, *smart healthcare*, entre outros (AL-FUQAHA et al., 2015).

A Camada de Negócios é responsável por gerenciar de maneira global as atividades e serviços da IoT, criando modelos de negócios. Tem como responsabilidade analisar, implementar, monitorar e desenvolver elementos relacionados ao sistemas de IoT. Segundo Al-Fuqaha et al. (2015) esta camada provê suporte para processos de tomadas de decisões baseados na análise de Big Data.

2.3 INFRAESTRUTURA

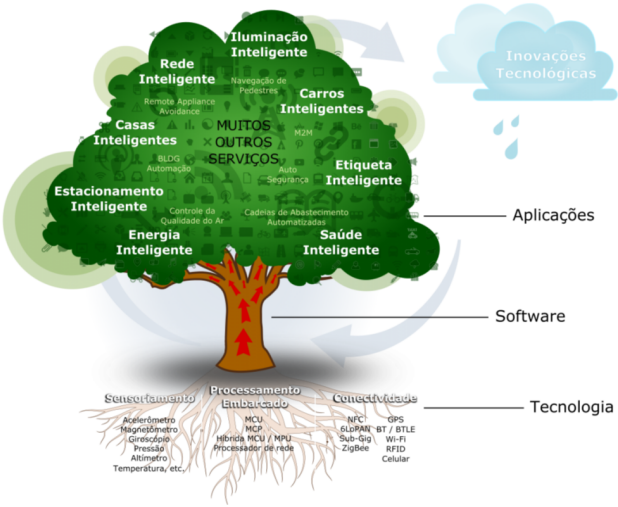
Para que a IoT seja uma realidade, Al-Fuqaha et al. (2015) apontam alguns elementos e tecnologias necessárias para o funcionamento da IoT, sendo eles identificação, sensoreamento, comunicação, processamento, serviços e semântica. A relação está ilustrada no Quadro 1. De maneira semelhante, a Freescale também relacionou as necessidades e tecnologias necessárias da IoT como ilustra a Figura 3 (KHAN et al., 2012).

Quadro 1 - Elementos IoT e Tecnologias Associadas

Elementos IoT		Exemplos
Identificação	**Nome**	EPC,Code
	Endereço	IPv4, IPv6
Sensoreamento		Sensores inteligentes, wearables, embarcados. Atuadores, RFID tags
Comunicação		RFID, NFC, UWB, Bluetooth, BLE, Z-Wave, WiFi, LTE-A
Processamento	**Hardware**	Arduino, Intel Galileo, Raspberry Pi, Beaglebone, Smart Phones
	Software	SO (Contiki, TinyOS, LiteOS, Android) Nuvem (Nimbts, Hadoop)
Serviços		Smart grid, smart homes, smart city, smart factors, etc.
Semântica		RDF, OWL, EXL

Fonte: Adaptado de (AL-FUQAHA et al., 2015).

Figura 3 – Tecnologia necessárias para IoT



Fonte: Adapatado de Al-Fuqaha et al. (2015)

O elemento de identificação (ID), como o nome sugere, é referente a identidade e também ao endereçamento únicos de cada dispositivo e serviços conectados na rede. Segundo Vasseur e Dunkels (2008), Kushalnagar, Montenegro e Schumacher (2007), Montenegro et al. (2007), há diversos métodos de identificação para sistemas IoT, tais como códigos de produtos eletrônicos (EPC do inglês *Eletronic Product Code*) e codificações ubíquas (uCode) (KOSHIZUKA; SAKAMURA, 2010). Para o endereçamento único na rede existem tecnologias como o protocolo IP (Protocolo de internet do inglês *Internet Protocol*) tanto as versões IPv4 como IPv6. A necessidade de obter tanto endereçamento como identificação únicas é justificada, segundo (AL-FUQAHA et al., 2015), pelo fato dos métodos de identificação, não abrangem uma identidade exclusiva em termos globais, tem-se a necessidade de um endereçamento único, por outro lado, objetos podem utilizar IP's públicos, necessitando métodos de identificação exclusiva para uma melhor distinção.

O elemento de sensoreamento esta relacionado com a camada de percepção, referente a arquitetura de cinco camadas, onde módulos sensores e atuadores possuem um núcleo de processamento como Arduino, Raspberry Pi, Beagle Bone, etc, capazes de disponibilizarem serviços,

tornando-se então *smart things*, objetos inteligentes, ao se conectarem na rede e transmitirem os dados coletados e processados (AL-FUQAHA et al., 2015).

Com respeito ao item comunicação, tem-se várias tecnologias e métodos para suprir tal necessidade, entre elas podem se destacar: WiFi (*Wireless Fidelity*), Bluetooth, IEEE 802.15.4. Algumas comunicações com fins específicos como RFID (*Radio-Frequency IDentification*), NFC (*Near Field Communication*) entre outras. A tecnologia RFID foi a pioneira em realizar a comunicação M2M (Máquina para Máquina do inglês: *Machine to Machine*). M2M é uma derivação do conceito de comunicação entre objetos da IoT, voltado para o setor industrial (WANT, 2006; AL-FUQAHA et al., 2015).

O elemento de processamento refere-se as unidades processadoras (microcontroladores, microprocessadores, FPGA's - *Field Programmable Gate Array*) integradas com software inteligentes de maneira a prover funcionalidades da IoT. Segundo (AL-FUQAHA et al., 2015), existem diversas plataformas de softwares para desenvolvimento das funcionalidades IoT, nestas plataformas encontram-se sistemas operacionais de tempo real (RTOS do inglês: *Real Time Operating System*), estes são de grande importância devido ao fato de executarem durante todo o tempo de execução dos dispositivos. Neste elemento também se encontram os processamentos em nuvem, possibilitando um comportamento ubíquo, onde a nuvem será responsável pelo processamento de dados mais complexos.

No elemento de Serviço, como o nome sugere, tem-se os serviços que a IoT pode fornecer. Neste contexto tem-se um vasto escopo de aplicações como visto na Figura 3. Algumas tecnologias como RFID e GPS (*Global Positioning System)*, permitem a integração desses serviços na IoT(AL-FUQAHA et al., 2015).

O elemento semântica segundo (BARNAGHI et al., 2012), é responsável pela interoperabilidade dos sistemas IoT, permitindo aos sistemas heterogêneos comunicarem entre si, transferindo dados não ambíguos. Para isto a semântica permite a extração de conhecimento entre máquinas, este conhecimento está associado ao descobrimento de dispositivos e serviços IoT na rede.

3 COMPUTAÇÃO AUTONÔMICA

Este capítulo aborda a definição da computação autonômica, suas propriedades, as propriedades auto's e a arquitetura base para outras arquiteturas de computação autonômica.

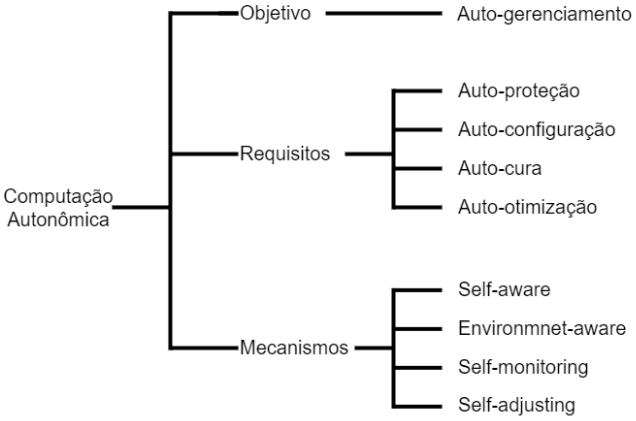
3.1 DEFINIÇÃO

O termo computação autonômica foi criado em 2001 pela IBM e segundo Horn (2001) a mesma tem como propósito gerenciar sistemas computacionais, sem a intervenção humana isto é, tecnologia gerenciando tecnologia. O termo autonômico é uma referência a biologia humana. O sistema nervoso monitora, de maneira autônoma, funções vitais sem a necessidade de um estímulo consciente da mente humana. De maneira análoga sistemas computacionais autonômicos identificariam problemas e necessidades do sistema e resolveriam utilizando pouca ou nenhuma intervenção humana (KEPHART, 2003; PARASHAR; HARIRI, 2005).

Sistemas autonômicos, de acordo com (MACEDO, 2012), são, em uma perspectiva, a evolução dos sistemas automáticos. Sistema automático tem como foco evitar a intervenção humana em suas operações, porém isto ocorre para uma faixa de entradas e processos bem definidos. Em contrapartida sistemas autonômicos devem ter a capacidade de extrapolar as saídas, isto é, se adaptar ao ambiente, mesmo que ocorra uma entrada não esperada pelo sistema, ou seja, deve ter a capacidade de continuar a operação e lidar com eventuais problemas.

O objetivo principal de um sistema autonômico é ser auto-gerenciado. Para alcançar esta meta, são necessárias quatro propriedades: auto-configuração, auto-cura, auto-proteção e auto-otimização (do inglês: *self-configuring, self-healing, self-protecting e self-optimizing*, respectivamente), como ilustra a Figura 4 (CORRÊA; CERQUEIRA, 2009).

Figura 4 – Propriedades gerais da computação autonômica



Fonte: Adaptado de Corrêa e Cerqueira (2009)

3.2 PROPRIEDADES DA COMPUTAÇÃO AUTONÔMICA

De acordo com Horn (2001), Corrêa e Cerqueira (2009) a definição das quatro propriedades são:

- Auto-cura: é a propriedade que tem como objetivo dar capacidade ao sistema de recuperar-se quando ocorre uma falha. Para isto deve habilmente identificar o tipo da falha e então, se possível, repara-lá. Isto deve ocorrer de maneira que os usuários sofram mínima interrupção, evitando também perdas de dados e atrasos significativos no processamento. Caso a recuperação não seja possível é desejável que o sistema continue funcionando sem o componente que veio a falhar. A auto-cura pode ser proativa onde antecipa a falha, através de mecanismos identificadores e realiza processos para prevenir que aconteça falhas.

- Auto-proteção: é a propriedade que tem como objetivo detectar, antecipar, e proteger contra ataques acidentais ou maliciosos. Os componentes de auto-proteção devem ser capazes de detectar comportamentos hostis e tomar decisões para tornar o sistema menos vulnerável.

- Auto-otimização: é a propriedade que tem como objetivo auto-ajustar o sistema para que o mesmo seja mais eficaz de acordo

com suas necessidades. O auto-ajuste pode fazer com que um dispositivo entre em modo de economia de energia, caso o nível da bateria esteja abaixo do aceitável. Outro exemplo é um sistema onde a temperatura está muito elevada, deve-se diminuir o processamento. Até mesmo aumentar a produtividade do sistema alocando mais recursos, caso haja a necessidade.

- Auto-configuração: é a propriedade que tem como objetivo dar a capacidade para o sistema se auto-organizar de maneira dinâmica a mudanças ocorridas no ambientes. Tais alterações podem ser oriundas devido ao funcionamentos das outras propriedades auto-cura, auto-proteção e auto-otimização, adição ou remoção de componentes do sistema, ou mudanças da sua própria característica.

Para que todos estes objetivos sejam alcançados o sistema autonômico deve possuir conhecimento de seu estado interno, *self-aware*, e as condições operacionais externas atuais, *environment-aware*. Através do auto-monitoramento (*self-monitoring*) deve detectar mudanças de contextos, assim as adaptações necessárias podem ser realizadas (*self-adjusting*). De maneira minuciosa, o sistema deve ter conhecimento de seus componentes, características de desempenho desejado, estado atual, a condição das conexões com outros sistemas e de seus recursos disponíveis (CORRÊA; CERQUEIRA, 2009).

3.3 ARQUITETURA DA COMPUTAÇÃO AUTONÔMICA

Segundo Corrêa e Cerqueira (2009) as arquiteturas para sistemas autonômicos oferecem, de maneira geral, soluções para automatizar o ciclo de gerência do sistema, como ilustra a Figura 5, o qual envolve três etapas:

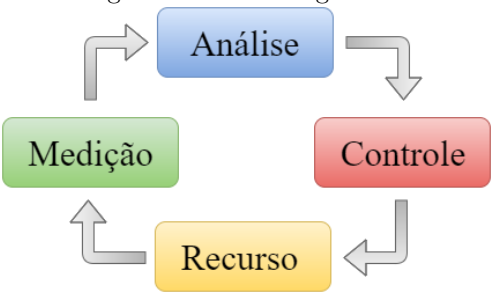
- Medição: coleta, filtra e agrega dados dos elementos gerenciados do sistema.

- Análise e decisão: com a obtenção dos dados, provenientes do monitoramento, examina-os e determina quais ações tomar.

- Controle e execução: executa o as ações e controla as mudanças sugeridas pela etapa anterior.

Muitas arquiteturas foram propostas para sistemas autonômicos (HUEBSCHER; MCCANN, 2008; DOBSON et al., 2006; ROSA; LOPES; RO-

Figura 5 – Ciclo de gerência



Fonte: Adaptado de Corrêa e Cerqueira (2009)

DRIGUES, 2006) e, segundo Macedo (2012), grande parte das arquiteturas são estruturadas de acordo com a arquitetura MAPE (Monitora, Analisa, Planeja, Executa) proposta pela IBM. Esta arquitetura é genérica para sistemas autônomos e serve como base para muitas adaptações e refinamentos.

A arquitetura MAPE foi proposta pela IBM em 2003, e foi adicionado uma letra K a sigla, que refere-se ao conhecimento, do Inglês, *Knowledge*, tornando-se assim MAPE-K e, conforme (TELES; SILVA; SILVA, 2011) este modelo esta sendo cada vez mais implementado. Sistemas autonômicos devem ser constituídos de elementos autonômicos todos interconectados.

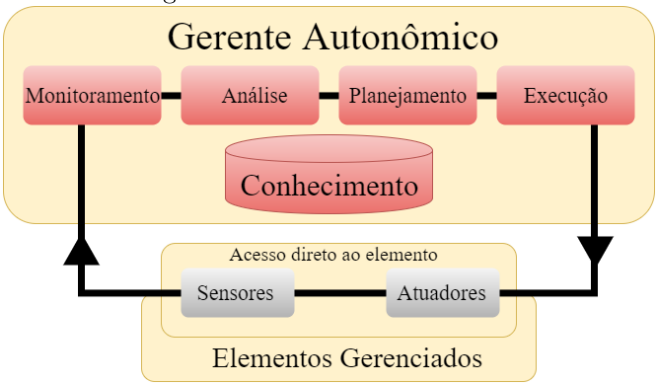
Elemento autonômico, como ilustra a Figura 6, é constituído de um ou mais elementos gerenciados, estes podem ser qualquer parte do sistema seja software ou hardware, e um único gerente autonômico, que será o responsável por efetuar os ciclos de controle propostos pela arquitetura MAPE. Para tal objetivo os elementos autonômicos devem possuir sensores e atuadores, assim são capazes de monitorar e atuar no meio em que estão instalados. Em algumas etapas, tomadas de decisões podem ocorrer, com mais frequência nos passos análise e planejamento, para isto foi adicionado uma base de conhecimento (STERRITT; BUSTARD, 2003; TELES; SILVA; SILVA, 2011).

Os dados coletados através dos sensores, são encaminhados ao primeiro passo, monitoramento, onde irá interpretar, processar e abstrair os dados para a fase seguinte, a análise. Nesta etapa, de acordo com a informação recebida, será avaliado a necessidade de mudança da configuração do elemento gerenciado. Para a avaliação podem ser empregadas diversas técnicas como inteligência artificial, estatística, teoria dos jogos, etc. A etapa planejamento é acionada quando na etapa

anterior, análise, é detectada a necessidade de uma modificação, caso contrário retorna-se para a fase de monitoramento. No passo de planejamento é identificado qual a melhor abordagem para a mudança no elemento gerenciado. A complexidade da mudança é proporcional a complexidade do sistema, este estudo será mais aprofundado na Seção 3.4. Após definido qual abordagem deve entrar em execução inicia-se a última etapa, onde será executado o plano apontado pelo passo anterior. Nesta última etapa constantemente é verificado se as instalações foram realizadas com sucesso (TELES; SILVA; SILVA, 2011; CORRÊA; CERQUEIRA, 2009).

Com relação a base de conhecimento, nela são armazenados informações do gerente autonômico e elemento gerenciado, como contexto e funcionamento dos mesmos, estado atual e previsões. Na base, também devem constar as políticas que regem o gerente autonômico, como é representado o conhecimento, para que facilite a comunicação entre gerentes autonômicos, para isto deve-se utilizar protocolos abertos (MACEDO, 2012).

Figura 6 – Elemento autonômico



Fonte: Adaptado de Kopiler (2007)

3.4 TOMADAS DE DECISÃO

Nas etapas análise e planejamento do gerente autônomo, haverá a necessidade de tomadas de decisões. Para este objetivo o elemento autônomo deve possuir, além da base de conhecimento, algoritmos que auxiliem nestas decisões. As escolhas destes algoritmos estão vincula-

dos ao grau de complexidade do sistema, e não do problema. Segundo Macedo (2012), alguns problemas complexos, podem ser solucionados utilizando métodos simples, sem a necessidade de algoritmos "inteligentes". Por exemplo, alguns sistemas requerem apenas uma mudanças de parâmetros, como trocar alguns valores. Assim uma regressão polinomial seria suficiente.

Para melhor entendimento das decisões e ter base para qual tipo de algoritmo/método utilizar, tem-se a necessidade de formalizar os problemas de decisão em dois tipos: imutáveis e mutáveis. Para tal objetivo, Macedo (2012) formaliza problemas de decisão com base na Equação 3.1.:

$$f = (i_0, i_1, ..., i_t) \longmapsto o_t \tag{3.1}$$

Onde f é função de resposta do sistema, possuindo um conjunto de entradas i e um conjunto de parâmetros de saídas o. Onde $t \in [0, \infty)$. Os valores de entradas e saídas podem ser tanto discretos quanto contínuos.

Problemas de decisão imutáveis são relativamente mais simples, onde para um mesmo conjunto de entradas sempre haverá a mesma saída, isto é, nem sempre para a mesma entrada haverá a mesma saída. Porém o contexto deve ser interpretado ou seja, para um conjunto de entradas o conjunto de saídas será o mesmo. Para formalizar o que são problemas de decisão imutáveis, e segundo Corrêa e Cerqueira (2009) estes devem atender a três condições:

- A quantidade de parâmetros de entradas e o seu domínio de valores não mudam com o tempo.
- A quantidade de parâmetros de saídas e o seus domínios de valores não mudam com o tempo.
- A função f é constante no tempo.

Devido a característica invariável do problema de decisão imutável, uma solução satisfatória pode ser proveniente de modelos tão simples quanto regressões lineares ou polinomiais, ou solução mais complexas como algoritmos genéticos, redes neurais, aprendizado de máquinas etc. A escolha do método irá depender das capacidades de quem irá implementá-las para identificar a função de resposta e da complexidade do problema.

Problemas de decisão mutáveis são aqueles que violam qualquer uma das três regras citadas anteriormente. Neste cenário os problemas de decisão devem se adaptar as novas regras de operações.

Problemas imutáveis podem ocorrer ao ser acrescentado um novo tipo de entrada, ou ser demandado novos tipos de respostas. Para este contexto, segundo Teles, Silva e Silva (2011), é aconselhável utilizar algorítimos ou métodos de aprendizagem de máquina ou inteligência artificial com aprendizado online. Assim capacita-se o sistema para se adaptar as mudanças do problema.

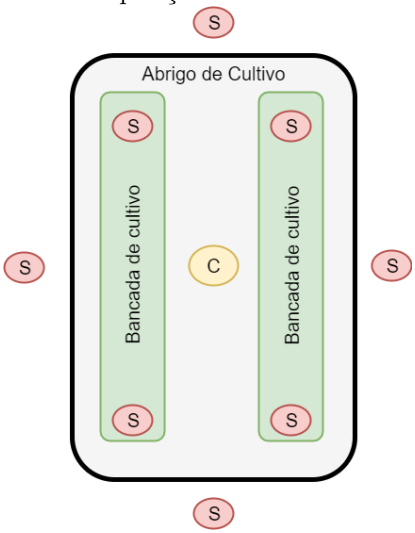
4 ARQUITETURA AUTONÔMICA PARA REDES DE SENSORIAMENTO EM ABRIGOS DE CULTIVO

Este capítulo descreve a arquitetura autonômica para redes de sensoriamento proposta neste trabalho e como ela está disposta em um ambiente de cultivo. Será apresentado a arquitetura da rede, características e seus módulos de hardware e de software.

4.1 ARQUITETURA AUTONÔMICA PARA ABRIGOS DE CULTIVO

A Figura 7 ilustra a disposição dos elementos da rede autonômica em um abrigo de cultivo. A central, representada pelo círculo ao meio com a letra 'C', deve estrar sempre posicionada em um ponto onde consiga receber e enviar mensagem para todos os sensores presentes na rede. Os círculos vermelhos com a letra 'S' em seu interior representam onde os sensores que devem estar posicionados nas áreas as quais pretende-se sensorear.

Figura 7 – Disposição dos sensores e central



Fonte: autoria própria

Utilizando a Figura 7 como exemplo, os sensores poderiam estar captando a luminosidade fora e dentro do ambiente, a central poderia fornecer estes dados para um sistema de controle de luminosidade, de maneira que o sistema de controle entre em ação de acordo com as informação fornecidas.

A arquitetura proposta consiste em 2 partes, o hardware sendo responsável por toda a parte física da arquitetura, sensores, microcontroladores, módulos de comunicação, etc, os quais devem prover condições suficiente para a execução dos softwares empregados como o gerente autonômico. O software que é responsável por toda a implementação dos conceitos de IoT e da computação autonômica, garantindo o funcionamento autônomo da rede.

4.1.1 Características de Hardware

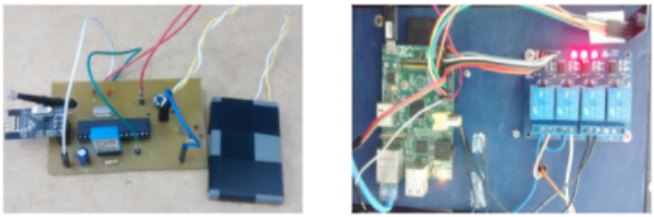
A parte física do sistema é composta por dispositivos de sensoreamento (módulos sensores) e uma central de processamento. Para tornar estes dispositivos em *smart things*, ambos dispõe de um módulo de comunicação e um microcontrolador similares aos da Figura 8. Além destes módulos os dispositivos de sensoreamento dispõe de um ou mais sensores para captar os dados do ambiente.

Os módulos sensores irão identificar-se ao entrarem na rede de comunicação, enviando uma mensagem para a central informando o tipo do módulo e quais serviços podem prover para o sistema do abrigo.

A central é composta por um microcontrolador com maior capacidade de processamento, comparado aos microcontroladores dos sensores. Este irá realizar o gerenciamento dos módulos sensores através de um gerente autonômico que irá aplicar os ciclos de gerência da computação autonômica, tratando os módulos sensores como elementos gerenciados.

Estes componentes citados até então se enquadram no contexto de "coisas" no âmbito de IoT, onde trocam informações e oferecem serviços em prol do sensoriamento do ambiente. A Figura 8 ilustra os módulos sensores e módulo central.

Figura 8 – Módulos: (a) Sensor; (b) Central



(a) (b)

4.1.2 Características de Software

A parte virtual do sistema conta com o gerente autonômico embarcado na central que irá constantemente monitorar os elementos gerenciados (módulos sensores), a fim de realizar as propriedades autonômicas de auto-configuração e de auto-cura. Sempre que o sistema não estiver operando da maneira correta a propriedade auto-cura deve ser acionada visando resolver o problema. Sempre que houver mudanças na configuração do sistema a propriedade de auto-configuração deve ser acionada, como estas operações serão realizadas dependerá das características do sistema

A central tem como responsabilidade prover e guardar todas as informações necessárias para o gerente autonômico executar de maneira correta, além de executar as funções regulares do sistema.

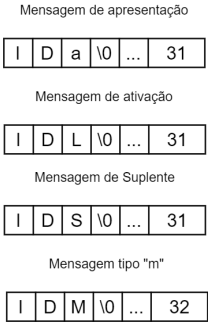
Todos os componentes, módulos sensores e central, devem utilizar o mesmo protocolo de comunicação para que seja possível a troca de informações entre os componentes, garantindo assim a comunicação entre os elementos da arquitetura, reforçando o conceito da IoT.

4.1.3 Funcionamento da Arquitetura autonômica para redes de sensoreamento em abrigo de cultivo

Como citado nas seções anteriores tem-se em mais baixo nível os elementos gerenciados, onde cada módulo terá uma identificação (ID) única, assim ao enviarem a mensagem de apresentação esta conterá o nome, tipo e estado dos mesmos possibilitando a central o reconhecimento de cada módulo. A central onde estará empregada o gerente autonômico, irá reconhecer e mapear os módulos e decidir se irá ou não

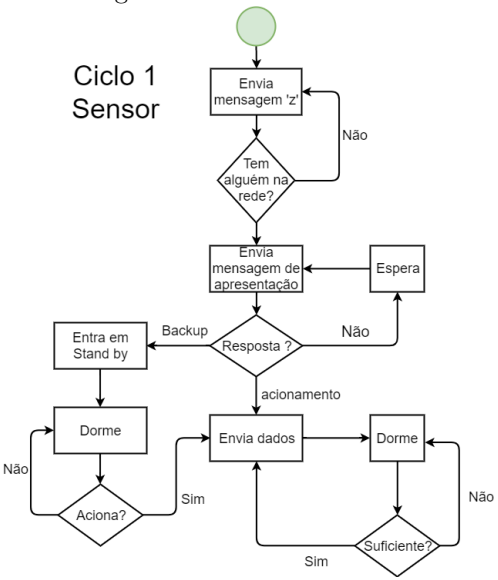
habilitá-los a entrar em execução. A mesma irá enviar uma mensagem contendo o ID do destinatário, e a ordem de execução ou *standy-by* como ilustra a Figura 9. Os módulos sensores ao receberem a resposta positiva da central começam a funcionar e enviar os dados de sensoriamento para a central, que irá armazenar e analisar as informações recebidas, mantendo um histórico dos dados para que o gerente autonômico possar fazer análises nestes dados. Os módulos sensores executam dois ciclos em paralelos, o primeiro refere-se ao funcionamento padrão de um módulo de sensoriamento. Neste primeiro ciclo o módulo irá se apresentar a central, esperar uma resposta positiva para começar a executar ou se tornar um módulo backup como ilustra o fluxograma da Figura 10. Já o segundo fluxo, mesmo sendo simples, é o que torna o dispositivo inteligente, pois neste ciclo o módulo de sensoreamento irá "conversar"(mensagem *'m'*) com a central sobre o seu desempenho, esperando como resposta se deve continuar em execução ou não como ilustra o fluxograma da Figura 11.

Figura 9 – Formatos das mensagens



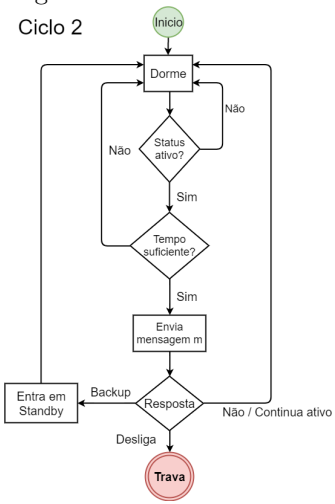
Fonte: autoria própria

Figura 10 – Fluxo 1 Sensor



Fonte: autoria própria

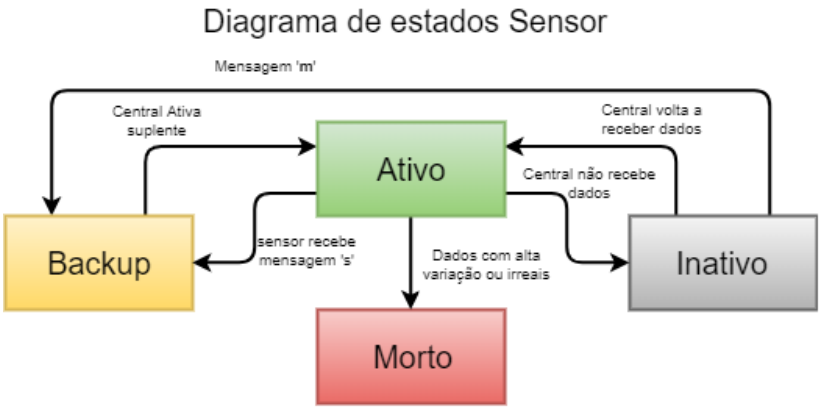
Figura 11 – Fluxo 2 Sensor



Fonte: autoria própria

A central é a parte principal da rede de sensoriamento, responsável por implementar o gerente autonômico e tornar a rede autônoma. Ela irá executar de acordo com algumas regras pré-estabelecidas como quantidade de sensores desejados, quais e quantos tipos de dados irão ser processados (temperatura, luz, etc) e tipo dos dados de sensoriamento. A central deverá atribuir estados (Ativo, Backup, Inativo e Morto) para os sensores que entrarem em contato com ela via mensagem de apresentação ou via a mensagem *'m'* ilustrada na Figura 11. O estado ativo representa sensor em funcionamento, Backup sensor suplentes, inativo sensor que há determinado tempo deixou de enviar dados e morto sensor com mal funcionamento. O gerente autonômico implementado na central que irá lidar com cada mudança de estado dos sensores. A Figura 12 Ilustra o digrama de mudança de estado dos sensores.

Figura 12 – Diagrama de estados



Fonte: autoria própria

Os sensores, em seu primeiro momento, ao conectarem-se na rede, possuem o estado ativo e enviam a mensagem de apresentação para a central, este pode mudar de acordo com a resposta. Caso a mensagem seja positiva o estado permanece o mesmo e começa a enviar dados de sensoriamento, caso contrário o sensor entra em estado de *backup* esperando ordens de execução. A resposta para a mensagem de apresentação somente poderá causar a mudança de estado de ativo para backup.

No estado de backup só há uma mudança de estado possível, de backup para ativo. Para ocorrer esta mudança o sensor no estado de backup envia periodicamente uma mensagem pedindo permissão para entrar em execução, ao receber resposta positiva a mudança de estado ocorre e o sensor passa a sensorear o ambiente.

A mudança de estado ativo para inativo ocorre quando as mensagens de sensoreamento do sensor não alcançarem mais a central ou ocorrer o desligamento do sensor. O inverso é possível acontecer, se o problema de recebimento de seus dados não existir mais, o sensor passa ao estado ativo novamente. Ainda há a possibilidade da mudança de inativo para backup, isto pode ocorrer quando a primeira mensagem de um sensor inativo chegar a central for de tipo *'m'* e a mesma dar a ordem para entrar em backup.

A resposta da central devido a uma mensagem do tipo *'m'* pode causar a mudança de estado inativo para ativo ou backup e de ativo para backup ou morto.

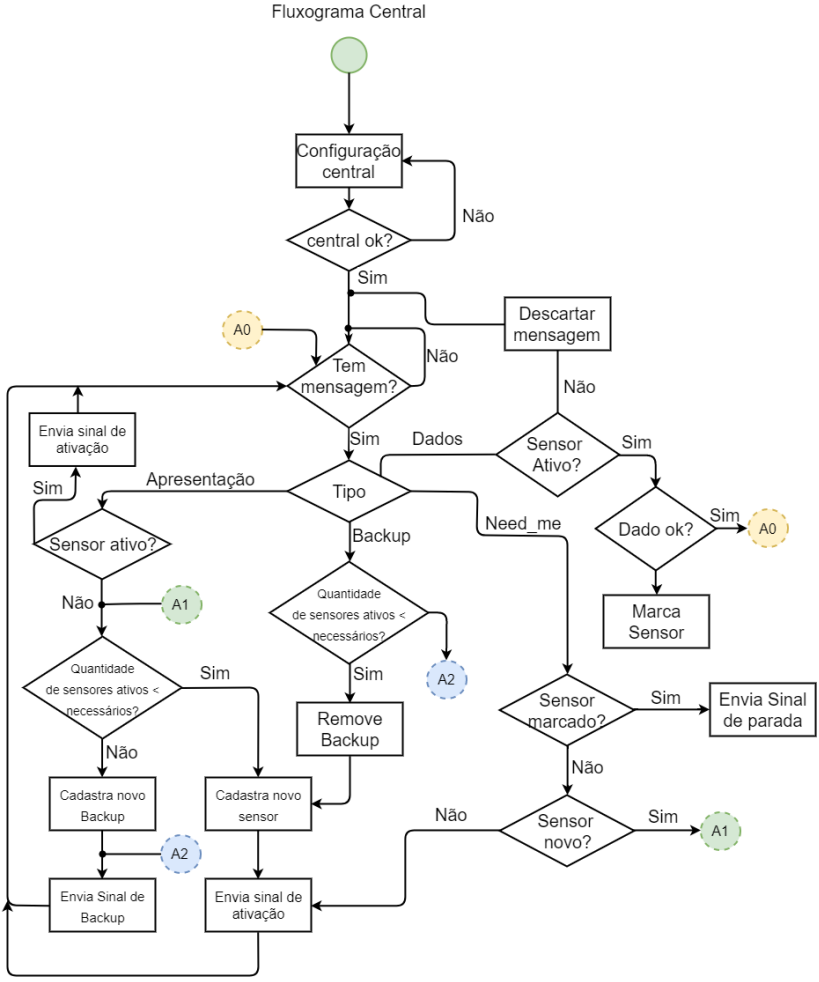
O sensor apenas entrará no estado morto caso esteja no estado ativo e envie dados com alta variação, não tendo consistências nos dados. Ou até mesmo dados irreais, exemplo luz ou umidade negativas. Ao adentrar neste estado o sensor irá travar e não irá mais fazer parte da rede. A única maneira de o sensor voltar a operar será através da ação humana.

Todas as mudanças de estado são responsáveis pelas ações do gerente autonômico o qual irá constantemente analisar os dados recebidos dos elementos gerenciados, a fim de acionar as propriedades de auto-configuração e de auto-cura. Caso ocorra alguma mudança no sistema, como a inserção de novos elementos gerenciados, a propriedade auto-configuração irá ser executada, ajustando o sistema para a nova configuração. Caso uma falha seja detectada pelo gerente autonômico, a propriedade auto-cura será acionada para que garanta a funcionalidade do sistema de sensoreamento, sem a perda de confiabilidade, ou seja, que o sistema execute de maneira correta mesmo com dados impróprios ou por queda de sensores.

Por exemplo, caso um módulo sensor venha a falhar, o sistema irá executar um protocolo de recuperação, detectando qual o tipo da falha e qual a melhor abordagem para a recuperação do sistema. Caso seja possível recuperar o módulo, ou acionar um módulo que esteja em *stand-by*, neste caso, apenas a propriedade auto-cura irá ser ativada. Porém se não for possível contornar a falha, ou seja, a propriedade auto-configuração deve ser acionada para que o sistema seja configurada para executar mesmo com a perda do sensor. As Figuras 10 e 11 acima e

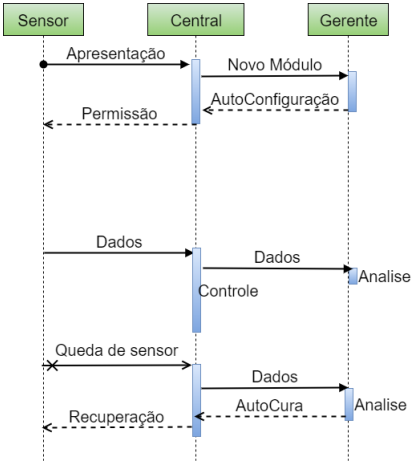
as Figuras 14 e 13 apresentam alguns exemplos de funcionamento do sistema.

Figura 13 – Fluxo Central



Fonte: autoria própria

Figura 14 – Diagrama de Sequência



Fonte: autoria própria

Desta maneira a arquitetura visa minimizar as ações humanas para gerenciar a rede, pois utiliza-se conceitos da IoT para o funcionamento e propriedades de computação autonômica para que ela esteja funcionando mesmo em condições adversas.

5 AVALIAÇÃO DO SISTEMA DE MONITORAMENTO

Este capítulo descreve os experimentos e os resultados obtidos com a arquitetura de sensoriamento autonômica proposta. Inicialmente o capítulo descreve a metodologia de avaliação empregada e em seguida os resultados obtidos com os experimentos realizados.

5.1 PROJETO DE EXPERIMENTAÇÃO

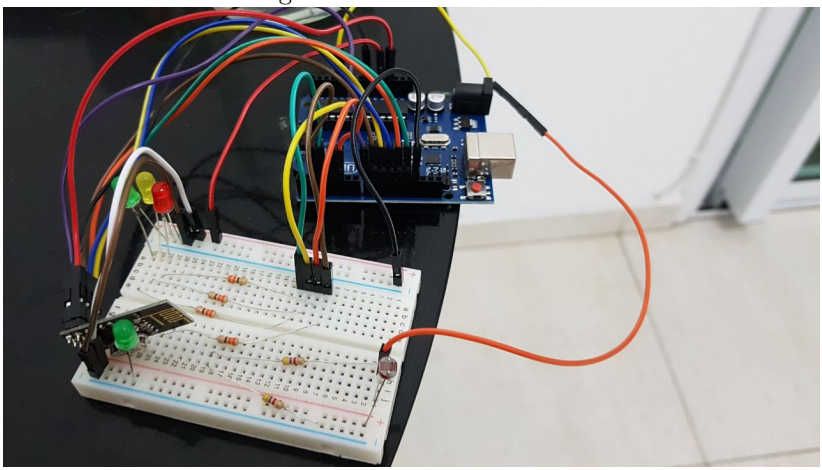
Para a implementação da elementos sensor e central presentes na arquitetura foram utilizados os seguintes componentes:

- 4 Arduinos Uno;
- Sensores de Luminosidade LDR;
- Módulos de transmissão sem fio nrf24l01;
- Led's vermelho, verde e amarelo;
- Resistores 330 Ω e 4,7k Ω;
- Jumpers diversos;
- 3 Protoboards de 400 furos.

O módulo central foi projetado com um Arduino Uno, um módulo rádio modelo nrf24l01 e um cabo usb para comunicação com o computador onde, através da comunicação serial da USB, enviará os dados da rede para o computador. O módulo sensor foi equipado com todos os itens descritos na lista acima. Os led's são utilizados para representar em qual estado o módulo se encontra (Ativo - verde, Backup - amarelo, Morto - vermelho), o sensor de luminosidade para adquirir dados do ambiente e por fim os jumpers e protoboards para fazer as conexões entre os equipamentos como ilustra a Figura 15.

Os softwares da central e do sensor foram desenvolvidos na liguem C, programando diretamente o microcontrolador, ATMEGA328P-PU. Optou-se por esta abordagem devido a existência de bibliotecas prontas para a comunicação sem fio para o módulo nrf24l01, melhor otimização do código e necessidade de controle das interrupções. A estrutura dos softwares seguem os fluxogramas das Figuras 10, 11 e 13 ilustradas no Capítulo 4.

Figura 15 – Módulo Sensor



Fonte: autoria própria

Para realizar os Ciclos dos sensores foi utilizado interrupção por tempo usando o *Watchdog timer* que foi configurado para acionar a interrupção a cada 8 segundos. Para o ciclo 1 quando o sensor estiver com o estado ativo irá enviar dados do ambiente (luz) para a central a cada interrupção e para o ciclo 2, a cada 40 segundos irá enviar a mensagem do tipo *'m'*. A cada 5 mensagens de dados o sensor irá enviar uma mensagem do tipo *'m'*. A frequência de envio de mensagens foi configurado desta maneira para que se pudesse observar o comportamento do sistema e o funcionamento da arquitetura de maneira mais prática e rápida.

Para que ocorressem as trocas e envios de mensagens entre a central e o sensor foram definidos alguns parâmetros e configurações:

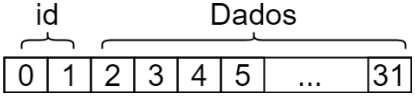
- Tamanho máximo da mensagem de 32 caracteres.
- A mensagem deve terminar com o caractere '\0'.
- Os id's dos sensores devem possuir 2 caracteres. Exemplo: "D0".
- O sensor, ao enviar mensagem para central, deve anexar o seu id nos dois primeiros caracteres da mensagem, preenchendo as duas primeiras posições, exemplo: "D0...\0".
- A central, ao enviar mensagem para um sensor, deve anexar o id do mesmo nos dois primeiros caracteres da mensagem. Exemplo "D0...\0".

- O sensor, ao enviar dados de sensoriamento, deve colocar o caractere '|' na terceira posição da mensagem antecedendo o dado de sensoriamento. Exemplo: "D0|56\0".
- Os dados de sensoriamento serão do tipo inteiro.
- Os tipos de mensagens da central, caracteres vindos após o id do sensor são.
  - Letra *'l'* acionamento de sensor.
  - Letra *'s'* colocar sensor em modo backup
  - Letra *'o'* desligar/travar sensor.
- Os tipos de mensagens do sensor são:
  - Letra *'a'* apresentação.
  - Letra *'m'* mensagem tipo m, a qual espera retorno da central avisando se deve continuar em ativação.
  - Letra *'n'* mensagem quando em estado de suplente esperando retorno da central se deve entrar em ativação.

A mensagem que o sensor envia em estado de suplente foi organizada desta maneira pois usualmente os sensores tem uma fonte de alimentação de energia limitada, assim o sensor não fica todo o tempo em ativação devido ao fato de entrar em modo de economia de energia após cada resposta vinda da central.

A Figura 16 ilustra o vetor de caracteres de 32 posições.

Figura 16 – Mensagem de 32 caracteres



Fonte: autoria própria

5.2 PROPRIEDADES AUTO-CONFIGURAÇÃO E AUTO-CURA.

A propriedade de auto-cura do experimento representa a capacidade do sistema em ativar os sensores suplentes quando algum sensor ativo passar a ter um comportamento atípico enviando dados fora da realidade do sistema (luz negativa, alta variação nos dados) ou pelo fato

de algum sensor ficar inativo. Já a propriedade de auto-configuração é acionada sempre que um sensor novo envia uma mensagem de apresentação ou do tipo *'m'* assim atribuindo o estado necessário para estes sensores ou quando a propriedade auto-cura é acionada, visto que sempre que esta última propriedade for acionada irá alterar a configuração do sistema, seja desativando um sensor ou ativando um módulo suplente.

Para identificar que um sensor entrou em inatividade a central utiliza o mesmo tipo de interrupção interna que o sensor, ou seja , a cada interrupção a central diminui uma unidade de um contador para cada sensor ativo, e, caso esse contador chegue a 0 o estado do sensor se torna inativo. Porém a cada mensagem que a central recebe do sensor este contador é incrementado. O valor deste contador, assim que um sensor se torna ativo, é 4, ou seja caso o sensor não envie nenhuma mensagem em um intervalo de 40 segundos, o mesmo se tornará inoperante, acionando a propriedade auto-cura caso haja um sensor suplente, e após isto a propriedade auto-configuração.

5.3 DESCRIÇÃO DOS EXPERIMENTOS REALIZADOS

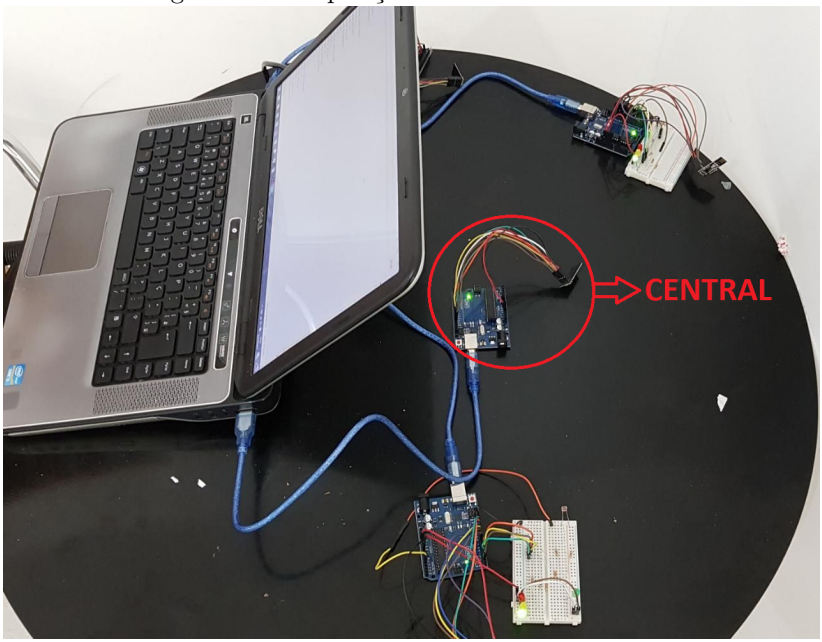
Para validar o funcionamento da arquitetura o sistema foi submetido e avaliado em 3 cenários. O cenário 1 simula uma rede que dispõe de um sensor em funcionamento e outro em estado de backup. O cenário 2 simula uma rede que dispõe de dois sensores em funcionamento e apenas um em estado de backup. E o cenário 3 um sensor em execução e outros dois em estado de backup.

Todos os cenários foram realizados em uma bancada de metal, onde o módulo central estava localizada entre os sensores, conforme ilustra a Figura 17, e conectado em uma porta USB de um computador por onde enviava as informações dos acontecimentos da rede. Para visualizar estas informações foi utilizado a IDE de programação do arduino, pois a mesma possui uma ferramenta capaz de monitorar as portas seriais como ilustra a Figura 18.

Para a comunicação estre os sensores utilizou-se uma adaptação de um dos protocolos utilizado em IoT, MQTT (*Message Queuing Telemetry Transport*), o qual realiza a comunicação através de um *broker* que seria um dispositivo, geralmente um servidor em nuvem, com papel de correio, onde todas as mensagem devem passar por ele e o mesmo irá encaminhar as mensagens aos outros dispositivos (AL-FUQAHA et al., 2015). Para este experimento o broker estava localizada na própria cen-

tral a qual ao receber as mensagens dos sensores do tipo apresentação (*'a'*), *'m'* e *'n'* encaminhava respostas para os mesmos.

Figura 17 – Disposição dos elementos da rede

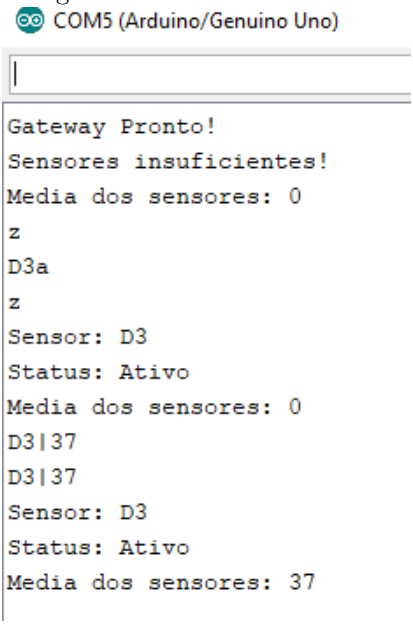


Fonte: autoria própria

Os experimentos realizados tiveram como objetivo testar a propriedade auto-cura, através da detecção de sensores inativos e o acionamento dos sensores suplentes, e a propriedade auto-configuração, após o acionamento da auto-cura quando detectava os sensores em estado inativo e acionava os sensores suplentes. Para tal objetivo conectavam-se os elementos na rede, esperava-se a central reconhecer todos os módulos, acionando a propriedade de auto-configuração organizando a rede de acordo com a quantidade de sensores necessários em estado ativo e backup, e por fim começar a operar. Após a rede estar configurada desligava-se um sensor ativo e através de um cronômetro, computava-se o tempo, em segundos, para a mudança do estado ativo para inativo referente ao sensor desligado propositalmente e em seguida computava-se o tempo de acionamento dos sensores suplentes. Para cada cenário foram realizados 5 testes, dos quais o melhor e o pior resultado, os que tiverem menos e mais tempo, respectivamente, foram descartados,

visando calcular a média de ativação das propriedades autonômicas.

Figura 18 – Atividade da rede



Fonte: autoria própria

Para o cenário 1 foram utilizados 2 sensores, 1 ativo e outro suplente, os resultados obtidos estão listados na Tabela 1.

Tabela 1 – Resultados do cenário 1

Cenário 1		
Teste	Notar inatividade	Acionar suplente após inatividade
1	40,41	8,32
2	40,21	5,49
3	41,68	4,28
4	42,04	1,99
5	40,67	1,97

Os testes 1 e 5 foram descartados, e a média obtida foi de 41,31 segundos para notar inatividade de sensor e 3,92 segundo para ativar o backup após inatividade

Para o cenário 2 foram utilizados 3 sensores, 2 ativos e 1 de suplente, sendo que os resultados obtidos estão na Tabela 2.

Tabela 2 – Resultados do cenário 2

Cenário 2		
Teste	Notar inatividade	Acionar suplente após inatividade
1	46,82	4,32
2	44,34	12,52
3	36,17	16,25
4	40,42	11,60
5	37,22	49,27

Os testes 1 e 5 foram descartados obtendo-se uma média de 40,31 segundos para notar inatividade de um sensor e 13,46 segundos para a ativação do módulo suplente.

Para o cenário 3 foram utilizados 3 sensores, 1 ativo e 2 suplentes, os resultado obtidos obtidos estão na Tabela 3.

Tabela 3 – Resultados do cenário 3

Cenário 3		
Teste	Notar inatividade	Acionar suplente após inatividade
1	36,45	8,69
2	40,01	7,18
3	38,28	infinito
4	34,32	8,72
5	13,13	8,16

Os testes 3 e 5 foram descartados, a média obtida para detecção de sensor inativo foi de 36,35 segundos e para o acionamento dos módulos suplente foram 8,2 segundos.

5.4 DESEMPENHO DA REDE AUTONÔMICA

A rede implementada com todas as características descritas nas subseções anteriores, teve o funcionamento muito similar ao esperado, conforme programado no início do capítulo, salvo o teste número 3 do Cenário 3, devido ao fato dos sensores de backup não conseguirem enviar suas mensagens para a central. Foi observado, porém não com-

putado, o tempo que a central levou para configurar a rede antes de serem realizados os testes. Percebeu-se algumas falhas no hardware dos sensores que estavam utilizando alimentação de uma fonte não proveniente da placa do Arduino Uno. Neste caso, era sempre necessário reiniciar o módulo após conectá-lo na fonte de energia. Observou-se que ao aumentar a taxa de transmissões de dados, no Cenário 2, a central perdia muitas informações tornando os testes com um maior número de sensores impossíveis de serem executados. Isto pode ser atribuído a capacidade de processamento da central e pela má otimização do código de recebimento de dados pelo módulo nrf24l01, pois ao receber as mensagens não conseguia processar a tempo os dados para que pudesse voltar a entrar no modo escuta.

6 CONSIDERAÇÕES FINAIS

A arquitetura autonômica implementada neste trabalho provou ser eficaz ao aplicar as propriedades de auto-cura e de auto-configuração da computação autonômica. Mesmo que de maneira limitada, manteve a rede operando mesmo sobre circunstâncias não favoráveis para todos os três cenários experimentados, tornando assim o gerenciamento da rede mais autônoma. Mesmo que ainda a arquitetura não remova o fator humano do gerenciamento da rede, pois para revitalizar um módulo sensor que esteja no estado morto é necessário a ação humana, a arquitetura garante o funcionamento da rede adaptando-se a algumas circunstâncias adversas. Assim, pode-se concluir que a arquitetura proposta irá diminuir a ação humana nos cuidados do sensoreamento, dos ambientes protegidos, e que irá fornecer dados confiáveis para qualquer sistema de controle que seja integrado no abrigo de cultivo, mesmo que alguma parte da rede não esteja funcionando corretamente.

Foi observado através dos experimentos realizados que os conceitos de *smart-things* está presente no funcionamento do sistema, onde os sensores são identificados pela central e que trocam informações para conseguir atingir um objetivo que neste caso é o sensoreamento de um ambiente protegido. Nota-se que os sensores são as "coisas"que fornecem serviços de monitoramento, e que a central é a consumidora destes serviços e que pode requisitar ou cancelar os serviços dinamicamente, sem a interação humana, de acordo com a necessidade. E, caso seja integrado um sistema de controle utilizando os conceitos de IoT, a central seria consumidora de serviços no ponto de vista a receber os dados dos sensores e seria uma prestadora de serviços fornecendo dados para o sistema de controle.

O protocolo MQTT mesmo que tenha sido adaptado para o experimento cumpriu o objetivo de gerar a comunicação entre os elementos da rede.

Mesmo que algumas premissas da arquitetura não foram cumpridas, como a central dispor de um hardware com maior capacidade de processamento, não influenciou negativamente o experimento. Infelizmente devido a esta quebra na premissa não foi possível analisar o comportamento da arquitetura para cenários mais complexos.

Com o experimento realizado foi possível perceber que o sensoreamento do abrigo está garantido mesmo que ocorram falhas em alguns módulos, deste modo o responsável pelo ambiente poderia dispor de mais tempo para agir no ambiente caso alguma falha ocorresse.

A arquitetura proposta além de fornecer mais tempo para agir durante uma falha para os responsáveis do abrigo, também garante um funcionamento uniforme devido a utilização de duas propriedades, auto-cura e auto-configuração, da computação autonômica.

Para os cenários e testes realizados a arquitetura se mostrou consistente, mesmo com alguns resultados negativos obtidos (teste 3, cenário 3), pode-se afirmar que os objetivos foram atingidos de maneira satisfatória, sobre as restrições adotadas.

6.1 PROPOSTA PARA TRABALHOS FUTUROS

São listadas nesta seção, algumas propostas para trabalhos futuros que visam melhorar e/ou estender a arquitetura proposta:

- Implementar todas as propriedades da computação autonômica;
- Implementar a arquitetura proposta em um ambiente de cultivo real e comparar com uma rede comum de sensoreamento;
- Utilizar outras características da IoT como a computação em nuvem;
- Desenvolver um sistema de controle para abrigo de cultivos utilizando a arquitetura proposta.

REFERÊNCIAS

ABRAH, P. B.; RODRIGUES, A.; PAGIUCA, L. G. Cultivo protegido. **Cepea - Esalq/Usp**, p. 38, 2014. Disponível em: <http://www.cepea.esalq.usp.br/hfbrasil/edicoes/132/full.pdf>.

ABREU, C. S. P. de; BASTOS, T. J. et al. Automação de abrigos de cultivo para culturas hidropônicas. 2015.

AL-FUQAHA, A. et al. Internet of things: A survey on enabling technologies, protocols, and applications. **IEEE Communications Surveys & Tutorials**, IEEE, v. 17, n. 4, p. 2347–2376, 2015.

ARORA, V. et al. Multi-representation based data processing architecture for iot applications. In: IEEE. **Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on**. [S.l.], 2017. p. 2234–2239.

ASHTON, K. That "internet of things"thing. **RFiD Journal**, v. 22, n. 7, p. 97–114, 2009.

BARNAGHI, P. et al. Semantics for the internet of things: early progress and back to the future. **International Journal on Semantic Web and Information Systems (IJSWIS)**, IGI Global, v. 8, n. 1, p. 1–21, 2012.

CORRÊA, S.; CERQUEIRA, R. Computação autônoma: Conceitos, infra-estruturas e soluções em sistemas distribuídos. **Anais do 27o. Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC'09)**, p. 151–198, 2009.

DOBSON, S. et al. A survey of autonomic communications. **ACM Transactions on Autonomous and Adaptive Systems (TAAS)**, ACM, v. 1, n. 2, p. 223–259, 2006.

DUTTA, S.; BILBAO-OSORIO, B. In: WORLD ECONOMIC FORUM. **The Global information technology report 2012: Living in a hyperconnected world**. Geneva, Switzerland, 2012.

FRIESS, P. **Internet of things: converging technologies for smart environments and integrated ecosystems**. Aalborg, Denmark: River Publishers, 2013.

GUBBI, J. et al. Internet of things (iot): A vision, architectural elements, and future directions. **Future Generation Computer Systems**, Elsevier, v. 29, n. 7, p. 1645–1660, 2013.

GUISELINI, C. et al. Manejo da cobertura de ambientes protegidos: radiação solar e seus efeitos na produção da gérbera. **Revista Brasileira de Engenharia Agrícola e Ambiental**, SciELO Brasil, v. 14, n. 6, p. 645–652, 2010.

HORN, P. Autonomic computing: Ibm\'s perspective on the state of information technology. IBM, 2001.

HUEBSCHER, M. C.; MCCANN, J. A. A survey of autonomic computing-degrees, models, and applications. **ACM Computing Surveys (CSUR)**, ACM, v. 40, n. 3, p. 7, 2008.

JAMES, R. et al. The internet of things: A study in hype, reality, disruption, and growth. **Raymond James US Research, Technology & Communications, Industry Report**, 2014.

JUNIOR, B. Estufas e casas de vegetação. **Revista casa da agricultura**, v. 14, n. 2, p. 01–22, 2011.

KEPHART, J. An architectural blueprint for autonomic computing. **IBM Publication**, 2003. Disponível em: <http://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf>.

KEPHART, J. O.; CHESS, D. M. The vision of autonomic computing. **Computer**, IEEE, v. 36, n. 1, p. 41–50, 2003.

KHAN, R. et al. Future internet: the internet of things architecture, possible applications and key challenges. In: IEEE. **Frontiers of Information Technology (FIT), 2012 10th International Conference on**. Islamabad, Pakistan, 2012. p. 257–260.

KOPILER, A. A. **Introdução à Computação Circulatória**. Technical Report, May, COPPE/UFRJ, 2007.

KOSHIZUKA, N.; SAKAMURA, K. Ubiquitous id: standards for ubiquitous computing and the internet of things. **IEEE Pervasive Computing**, v. 4, n. 9, p. 98–101, 2010.

KUSHALNAGAR, N.; MONTENEGRO, G.; SCHUMACHER, C. **IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals**. Microsoft Corporation, 2007.

MACEDO, D. F. **Computação Autonômica**. Technical Report, UFMG, 2012.

MARANGONI, V. H.; SOUZA, P. S. de; MOREIRA, H. R. Automação de estufas agrícolas utilizando sensores e arduino. In: **6ª Jornada Científica e Tecnológica 3º Simpósio da Pós-Graduação**. Pouso Alegre, MG, Brasil: IFSULDEMINAS, 2014.

MONTENEGRO, G. et al. **Transmission of IPv6 Packets over IEEE 802.15.4 Networks**. Microsoft Corporation, sep 2007. 1–30 p. Disponível em: <https://www.rfc-editor.org/info/rfc4944>.

NUNES, D. S.; ZHANG, P.; SILVA, J. S. A Survey on Human-in-the-Loop Applications Towards an Internet of All. v. 17, n. 2, p. 944–965, 2015.

PARASHAR, M.; HARIRI, S. Autonomic computing: An overview. In: **Unconventional Programming Paradigms**. Berlin Heidelberg: Springer, 2005. p. 257–269.

PURQUERIO, L. F. V.; TIVELLI, S. Manejo do ambiente em cultivo protegido. **Instituto Agronômico de Campinas IAC, Centro de Horticultura. Campinas, SP**, 2006.

ROSA, L.; LOPES, A.; RODRIGUES, L. Policy-driven adaptation of protocol stacks. In: IEEE. **International Conference on Autonomic and Autonomous Systems (ICAS'06)**. IEEE, 2006. p. 5–5.

STERRITT, R.; BUSTARD, D. W. Towards an autonomic computing environment. IEEE Computer Society, 2003.

TELES, A. S.; SILVA, F. J. da; SILVA, Z. A. Computação autonômica aplicada a segurança de redes. 2011.

UZOCHUKWU, G. A. et al. **Proceedings of the 2013 National Conference on Advances in Environmental Science and Technology**. Greensboro, EUA: Springer, 2015.

VASSEUR, J.; DUNKELS, A. Ip for smart objects. **IPSO Alliance, White paper**, v. 1, 2008.

VERMESAN, O.; FRIESS, P. **Internet of things-from research and innovation to market deployment**. Aalborg, Denmark: River Publishers, 2014.

VIDA, J. B. et al. Manejo de doenÃ§as de plantas em cultivo protegido. **Fitopatologia Brasileira**, SciELO Brasil, v. 29, n. 4, p. 355–372, 2004.

WANT, R. An introduction to rfid technology. **IEEE Pervasive Computing**, IEEE, v. 5, n. 1, p. 25–33, 2006.

YANG, Z. et al. Study and application on the architecture and key technologies for iot. In: IEEE. **Multimedia Technology (ICMT), 2011 International Conference on**. Tai Yuan, China, 2011. p. 747–751.

YUN, M.; YUXIN, B. Research on the architecture and key technology of internet of things (iot) applied on smart grid. In: IEEE. **Advances in Energy Engineering (ICAEE), 2010 International Conference on**. Beijing, 2010. p. 69–72.